1

Introducing user and task analysis for interface design

The interface is what users see and work with to use a product. Interfaces can help or hinder, be effective or ineffective, as figure 1-1 shows. This book is about how to get and use the information you need to design helpful, effective interfaces.

Designing an effective interface doesn't happen by chance. Good design happens only when designers understand people as well as technology. Good design happens only when designers understand who will be using their product and the personal characteristics, habits of mind, physical capabilities, and limitations those users bring to their tasks. Good design happens only when designers understand what users are trying to accomplish—the users' goals and tasks—and how the users think about their tasks—the users' conceptual model of the work and the tools. Good design happens only when designers understand the circumstances under which users must work and how users as groups collaborate to accomplish a goal.

And good design is important. It makes users happy and productive, increasing customer satisfaction, one of the major goals for most companies. It increases efficiency and decreases support calls, thus making and saving money for companies.

Figure 1-1 Interfaces between product and user can help or hinder the user.

Cemex, a cement company in Guadalajara, Mexico, wanted to improve the speed and reliability of deliveries to improve customer satisfaction and internal efficiency. They set out to develop a new way for dispatchers to communicate with the drivers of cement trucks. Before they designed the new system, the designers held "conversations" with the dispatchers, and they watched the drivers. They learned what these people needed to do their jobs well and proudly. They then built those conversations into the design of the system and its interface. They succeeded. On-time deliveries of cement in Guadalajara, Mexico, went from 34.4% to 98.15% (Katel 1997).

When Federal Express wanted to redesign its ground operations manuals, JoAnn's team visited locations around the world, observing employees solving problems, using manuals on occasion, and often expressing frustration at how difficult it was for them to find the information they needed. The site visits led to a usability study to identify the details of the problems (organization, language, indexing, lack of examples, and so on). Following the predesign work, JoAnn's team and technical communicators at Federal Express reorganized and rewrote the manuals. They succeeded. Time to search for information in the manuals decreased by more than a third, resulting in a potential savings of more than $24 million per year in North America alone through reduced search times, better adherence to standards, and less need to call for help (Hackos 1995).

In both of our example stories, the design teams spent time observing and talking with users, listening to the people who actually did the work, watching how they worked, understanding how they performed tasks, and learning what the problems were by seeing them. They took that information back and analyzed it. They then created prototype designs and tested the prototypes with users until they met the usability goals of the projects. In fact, they used many of the techniques that we cover in this book.

Despite success stories like these, we all know that many products, especially software applications, are not easy to use. In large part, that's because the interfaces don't make sense to users who struggle to find what they need and to figure out what to do and how to do it. Tasks that were easy to do by hand or with older systems sometimes take more time and effort using a computer, a graphical user interface (GUI), or a new product design. As Tom Landauer (1995) has shown, productivity gains promised for computers in the workplace and the home have not been achieved.

There are many reasons for the problems that people have using interfaces. One is the pressure on designers to add more and more features. But the main reason for the problems is that products are too often developed with the focus on technology only and not on users. The heart of the problem is a lack of a firsthand, carefully considered understanding of the users, their tasks, and their environments.

Time and time again, each of us has been asked to rescue unusable designs, only to find that no one on the design team has ever observed a user doing the tasks for which they were designing. They were too busy to visit and observe users. Or if they did, they didn't know how to observe effectively or what questions to ask. Or they didn't know how to incorporate what they had learned into the design of the interface.

When designers visit users in their workplaces and homes to analyze who the users are, the work they do, and where they do it, the designs that result seem like magic. But it's not magic. It's hard work, good communication, analysis, and design techniques, and, yes, some vision and creativity, but vision and creativity based on knowledge.

What is this book about?

This book tells you how to get started, at the predesign stage, on the path toward designing a usable interface. We start out with the basics, explaining what you need to know about users, tasks, and environments. Then we take you through a step-by-step process for planning and conducting site visits. We include advice on how to be a good observer and listener and how to conduct an effective interview. Finally, we explain how to analyze the information you've gathered and turn it into

interface designs. We help you move through basic design steps until you're ready to create and test design prototypes. We also discuss the importance of including strategies and designs for documentation and training as part of your interface.

We've deliberately taken an eclectic approach in this book, introducing you to many different techniques and variations on techniques for data gathering, for analysis, and for moving from analysis to design. What all of our techniques have in common is direct involvement with users and a focus on the work that users do. What they all have in common is that they are ways of doing user-centered design, by having this information before design and using it to keep everyone on the design team focused on the users and their work.

Other books and book chapters present their authors' particular methodologies only (for example, Beyer and Holtzblatt 1997; Dayton, McFarland, and Kramer 1998). We have included several of these techniques with credits to their authors among the many that we present for your consideration. We also discuss other techniques that we have used in both our practices as well as some that we have learned about from presentations at conferences and workshops.

We expect that you will adopt and adapt techniques from this book for your projects. You are likely to pick and choose from among the techniques and to formulate your own methods that are combinations of a number of useful techniques. [For example, that's what the Hiser Group, a consulting firm in Melbourne, Australia, did. They work with a methodology that they call the "Element Tool Kit." It is a collection of techniques, all of which are covered in this book, that they have selected for data gathering, analysis, design, and evaluation. They have found that the techniques they've incorporated into their methodology work well for their clients and the projects they deal with (Hiser 1997)].

You are also likely to find, as we have, that different techniques and combinations of techniques are needed on different projects. The circumstances may be different. The issues that you need to learn about may be different. The time you have to gather information, analyze it, and use it may be different. Some techniques may fit well with your company's corporate culture and others may not. We've tried throughout this book to suggest how to select appropriately among techniques for different situations. So take in the techniques we discuss in this book with a view not only to what you want to do now, but also with a view to what might be helpful in other projects in the future.

What we do not include in this book is information on how to do graphic design in the development of a GUI or how to apply the design standards promulgated by the operating system developers—Windows, Apple Macintosh, X-Windows, OS/2, and other standards. Good advice on design and books on these standards already exist. You'll find references for many resources on design and standards when you get to

chapter 12, which is about turning analysis into design. In the appendixes, you'll find a checklist of design standards that JoAnn created as an amalgam of the platform-specific standards. But right now we hope you are ready to think about interfaces, not from the view of the operating system, but from the users' view of the work they need to do.

What is interface design?

All the products that we mention in this book involve interactions between humans and systems. The systems are sometimes computer based, sometimes document based. The first story we told in this chapter is about the interface to a software application for dispatching cement trucks. The second story we told is about the interface (organization and page layout) of a set of manuals that is currently delivered on paper but will eventually be delivered electronically. We could also tell stories, and we do, about employee benefits handbooks, cellular phones, hardware and software for home use, office software, online help, and many others. All of them involve designing effective interfaces to make the products usable.

Interfaces occur in everything you work with, including

- the controls on a hardware product
- the labels and signs on the hardware
- small liquid crystal displays on machines of all sorts
- the screens for software applications on mainframe terminals
- the screens for software applications on personal computers running operating systems such as Windows, OS/2, DOS, Macintosh, UNIX, and others
- the pages of a Web site
- help systems and online and paper manuals
- embedded tutorials and other types of performance support
- the page layouts of paper forms or other documents

An interface is the bridge between the world of the product or system and the world of the users. It is the means by which the users interact with the product to achieve their goals. It is the means by which the system reveals itself to the users and behaves in relation to the users' needs.

What makes an interface usable?

To be usable, an interface must let the people who use the product (users), working in their own physical, social, and cultural environments, accomplish their goals and tasks effectively and efficiently. To be usable, an interface must also be *perceived* as usable by those who must use it or choose to use it. They must be pleased, made comfortable, even amazed by how effectively their goals are supported by your design. In the best case, they will be oblivious to the design—it simply works so well that they don't notice it. The truly usable interface is transparent to the work the user is trying to accomplish. (See figure 1-2.)

Figure 1-2 A user interacting with a usable interface may be unaware that an interface is there at all.

Usable interfaces have certain characteristics in common:

- They reflect the workflows that are familiar or comfortable.
- They support the users' learning styles.
- They are compatible in the users' working environment.
- They encompass a design concept (a metaphor or idiom) that is familiar to the users.
- They have a consistency of presentation (layout, icons, interactions) that makes them appear reliable and easy to learn.

- They use language and illustrations that are familiar to the users or easy to learn.

In short, usable interfaces fit in, simply and elegantly, with the users' life and work needs. If not immediately obvious to the users, they are quickly learnable. As we've said, such usable interfaces rarely happen by chance.

What is user and task analysis?

If we look at the history of successful designs, we often find designers who have developed remarkable insights into the way people work and learn. Those insights most often come from a close connection between designers and users. Designers who spend time with users, observing how they work, understanding who they are, testing design concepts and prototypes, are most likely to be successful in creating interfaces that are a delight to use.

The name we and others in the field of user-centered design give to these processes of interaction between designers and users is user and task analysis.

User and task analysis is the process of learning about ordinary users by observing them in action. It is different from asking them questions in focus groups outside the users' typical environments and away from their work. It is different from talking with expert users or managers who may claim to speak for ordinary users but often unknowingly misrepresent them. Such "user representatives" may know how users are supposed to work, or they remember how they did the work when they were users, but they often do not know what really happens in the users' world today.

In fact, experience has shown that users themselves do not know how to articulate what they do, especially if they are very familiar with the tasks they perform. If you invite users to focus groups or meet with them in conference rooms, they will tell you about what they do. But when you watch them performing their tasks, you will learn that their testimony is often incomplete and inaccurate. Users, like all of us, leave out activities that they don't even notice they're doing. They emphasize activities that they find difficult or boring or particularly exciting to the exclusion of ordinary activities. They report on what they believe to be true, not necessarily what is.

Only by observing users and by probing for more understanding in the context of their work will you get the information that will surprise you and make you rethink your design ideas. That means using site visits to do user and task analysis.

User and task analysis, as described here, focuses on understanding deeply how users perform their tasks today. That understanding includes

- what users' goals are; what they are trying to achieve
- what users actually do to achieve those goals
- what personal, social, and cultural characteristics the users bring to the tasks
- how users are influenced by their physical environment
- how users' previous knowledge and experience influence how they think about their work and the workflow they follow to perform their tasks
- what users value most that will make a new interface be a delight for them (speed? accuracy? help in recovering from errors? human contact? fun? a challenge?)

We have chosen to define user and task analysis broadly, encompassing some specific formal methods like constructing flowcharts or hierarchy diagrams of tasks, but primarily emphasizing the more qualitative and informal methods of watching, listening carefully, and probing for more understanding. User and task analysis of the sort we hope you will learn from this book requires gaining access to users or potential users of products from a wide variety of backgrounds and roles. It encourages the participation of many members of the development team, including those who decide what the software and hardware will allow users to do, those who create the software and hardware interfaces, those who write help systems and other documentation, those who design Web sites, and those who design instruction, whether delivered in the classroom or through electronic means.

Recent experience, especially in the design of application software, has shown that the qualitative methods emphasized in this book result in information that is most useful to interface design.

When should you do user and task analysis?

The major emphasis in this book is on user and task analysis before design begins, even before analysis begins. Successful design comes from a basis in direct observations of, not assumptions about, the users or potential users of your product. We have learned, and we hope we can impress on you that none of us, not even the most experienced designers among us, know enough about how particular users work and think to design using sheer analytical power alone. We have learned that interacting with users in their actual environments, and performing user and task analysis, is essential to designing usable products. That means we might as well start with the users rather than waste our creative energies on "neat" designs that don't and won't ever make sense to the users. It's amazing how many clever, and

costly, designs have been thrown out during usability testing or, worse yet, shelved or completely redesigned after they've been released to the unwitting users.

We've also learned that, although there are some general guidelines for good design, such as "Speak the user's language," and "Be consistent," each product is unique. Although there is general knowledge about users, such as "They are not blank slates; they always bring the baggage of all their prior experiences," each set of users is different. Although doing user and task analysis for one product may give you insights that you can carry over to other projects, you cannot rely exclusively on prior work. For each new product, each new design, each new set of users, each expansion into another market or another culture, you have to do a new user and task analysis.

We do not preclude the thought that some user and task analysis often takes place later in the development life cycle to prepare documentation and training to support a new or redesigned interface. Nor do we preclude gathering user and task information later in the development life cycle through usability testing and alpha and beta site product introduction. If you continue to observe users during the development life cycle, you will continue to gather new information.

The only problem with these late-cycle activities is that they are less likely to influence the direction of the interface design. The later in the life cycle that fundamental design problems are recognized, the more time consuming and expensive the changes will be to make. In fact, we know that even when seemingly minor design flaws are recognized, they are often not corrected in order to meet a predetermined schedule or stay within a budget. So instead of making costly changes, or trying to patch up inadequate interface designs with help, documentation, and training, we urge you to do it right the first time. The information you learn later should be for minor refinements not major surprises.

A beta site visit or a usability test right before final product release should not be the first time that you observe users interacting with your interface design. Therefore, we recommend strongly that user and task analysis occur early, before design decisions have been made and so that they can be incorporated into early design ideas. Figure 1-3 is a flowchart showing the most effective placement of user and task analysis in the interface-design process.

By including user and task analysis in the predesign stages of the life cycle, you have the opportunity to create inexpensive design models and test them with users before you have spent much time implementing the design into the technical environment. You also have the opportunity to ensure that the design of the underlying structure of the software promotes the efficient interaction of users and interface objects.

User and task analysis also fits well into the whole system development life cycle, saving time even for those developing the database model or creating the underlying system objects. You will find this to be the case because user and task

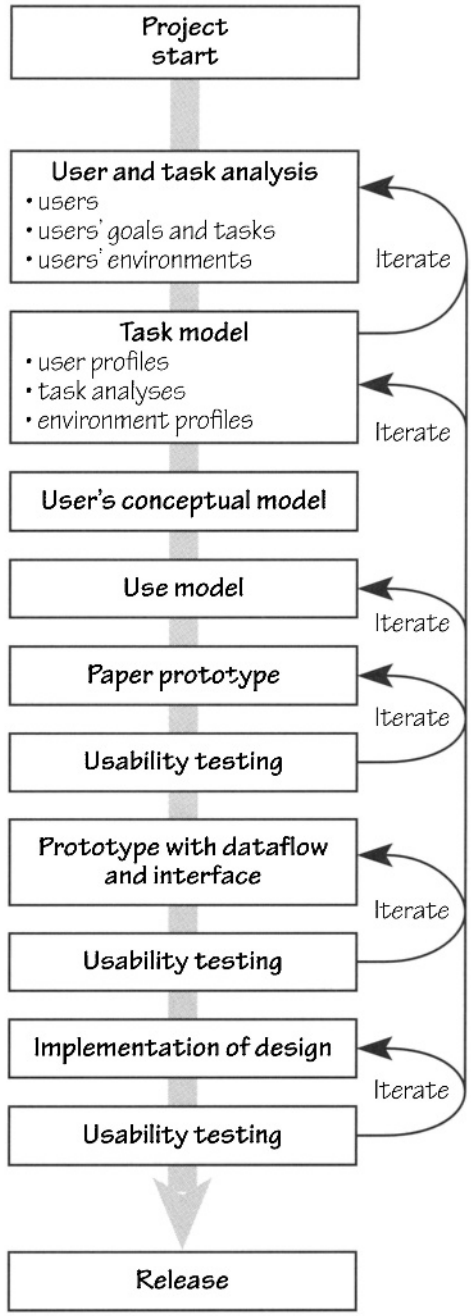

Figure 1-3 The relationship of user and task analysis to the interface-development life cycle.

analysis, as practiced following the methods in this book, will decrease the possibility that you will begin the underlying design of a system based on faulty information from users. We have all experienced the design delays and cost overruns that occur when we move forward with a design, only to have users tell us that the design represents "what they said" but not "what they meant." User and task analysis tells us that we have to gather information about what the users need from observing them at work, rather than only listening to their reports of what they think they want.

Figure 1-4 shows how user and task analysis figures into both the design of the interface and design of the underlying system.

Why do user and task analysis at all?

We recognize that user and task analysis has not been an integral part of interface design and product development. At the same time, we know that poor design takes an enormous toll on productivity and increases the cost of maintaining a product. Companies today pay double or more for poorly designed interfaces. First, they pay for the initial development costs; then they pay for the support costs when users call for help. They pay for field maintenance when users conclude that a usability problem is a product defect. They also pay the cost of making changes after the product is released.

Customer organizations also pay dearly for poorly designed interfaces. They pay when employees find new products difficult to learn. They pay when people make mistakes that have to be corrected. They pay when employees are unhappy and decide to quit rather than put up with a new interface. They pay when employees spend 20, 30, even 40% of their time asking each other how to do a task. They pay when all the employees settle on a method of performing the task that turns out to be inefficient, illegal, or just plain wrong.

Companies also pay for poor design in decreased sales. We know that it's relatively easy to sell to the early adopters, making initial sales to people who always buy the latest technology. But remember, no one ever had more than one pet rock. In the later stages of the sales cycle, customers rely increasingly on word-of-mouth recommendations. When the word-of-mouth is that your product is unusable or costs enormous sums for training and support, sales may be adversely affected.

User and task analysis, followed by other usability activities, can change all that, as the following story illustrates.

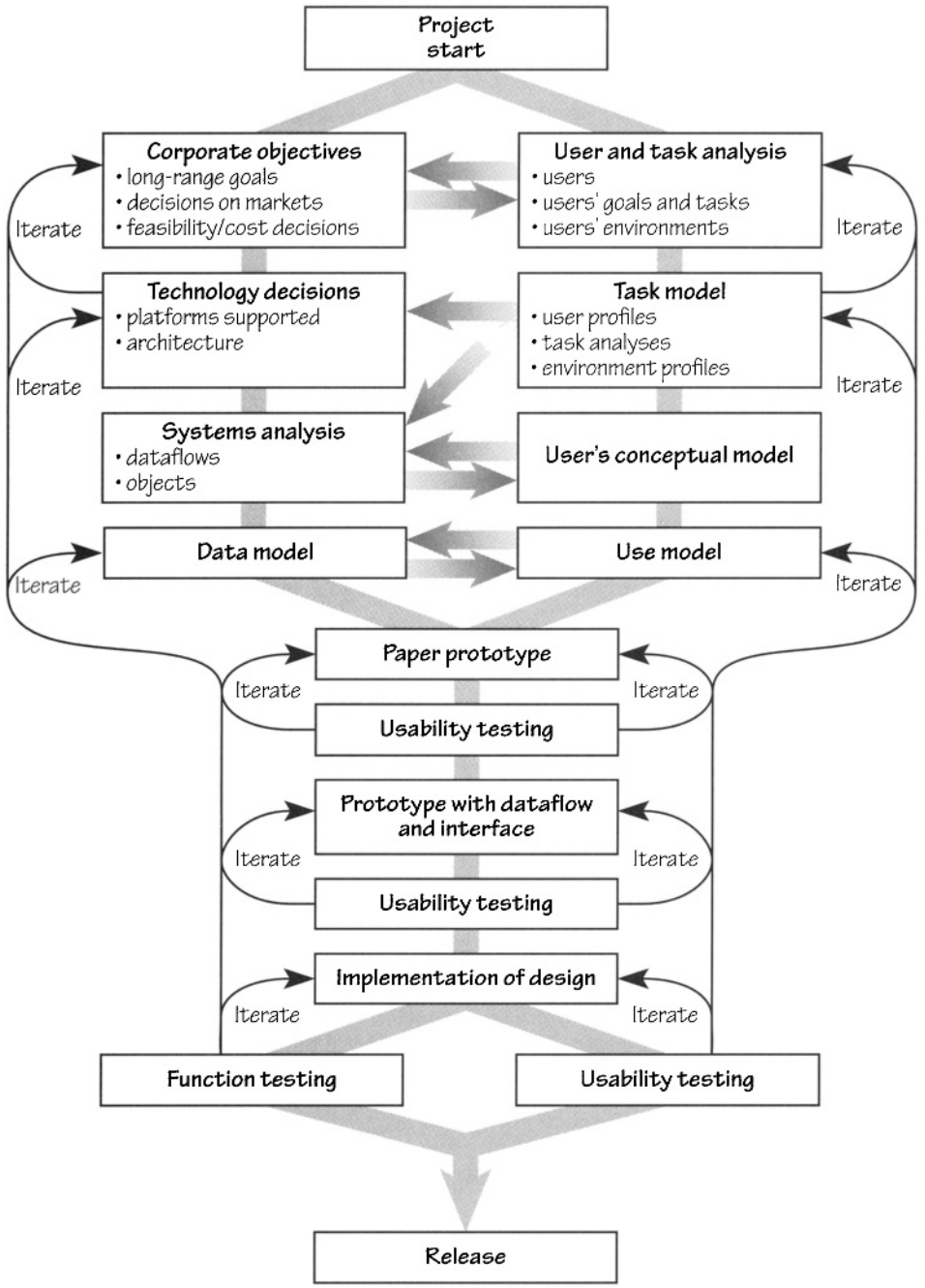

Figure 1-4 The relationship of user and task analysis to both the interface and the underlying system development.

The first version of Digital Equipment Corporation's RALLY software had disappointing sales. The company had to decide whether to scrap the idea or try again. The second time around they let the usability people take the lead. Dennis Wixon and colleagues observed and talked with users in the users' workplaces before doing any design. They used their techniques of contextual inquiry and affinity diagramming to do user and task analysis. They created prototypes and had users try them out as the design team observed and listened and talked with the users. They iterated the prototypes, taking into account what they'd learned from users, making changes, and having other users try out the changed version. When the second version of RALLY was released, it had excellent sales. Sales rose by 80%, some 30% to 60% above expectations (Wixon and Jones 1996).

Why isn't this done all the time already?

You, no doubt, have heard all the objections. You may even have voiced some of them yourself:

- Marketing already knows the users.
- The product is new—there aren't any users to observe.
- Our users are all too different—we can't possibly visit all of them.
- We don't have enough time in the schedule.
- We don't have enough money in the budget.
- One of the members of the development team was a user for 25 years.
- We really have some great design ideas already we don't need users to question what we're doing.
- Users don't know anything about design.
- Designers should design, not users.
- We are users ourselves—we know what we need.
- We're reengineering the process anyway—what can the current users tell us?
- We've never done user and task analysis we don't know how to do it.
- We've been holding focus groups and the users attend.
- We did a survey of the users and asked them what they wanted.

You will read more about these objections and how to counter them in chapter 5. However, remember that site visits can be scheduled and made quickly, that visiting a few users is infinitely better than visiting none, that a few pointed observations and quotations will often open the eyes of the most recalcitrant managers. The key, of course, is success. Try site visits and careful, effective observation on a small scale at first. See what you learn and how much it influences your thinking. Test your new design ideas with users and see what happens. And measure, measure, measure. Record the successes, calculate the cost savings, and analyze and redesign the failures.

Where does user and task analysis come from?

We don't claim to have invented any of the techniques that we present in this book. Some we have adapted and refined ourselves from work in other disciplines. Some we have learned, as we said earlier, from other usability specialists, and we've tried to give credit where due. Many of those techniques, we suspect, are also adapted and refined from other disciplines.

Usability is a relatively new field that has attracted people with a wide variety of backgrounds. People have come to it from anthropology and ethnography, cognitive psychology, computer programming, document design and technical communication, English and rhetoric, human factors, graphic design, instructional systems design, market research, scientific management, and systems engineering, as well as many other disciplines that we have not mentioned. Many of these disciplines have influenced the practice that we describe in this book. We should take a moment here to acknowledge some of the antecedents of the practice and also to explain how what we are talking about in this book differs from those antecedents.

Anthropology and ethnography

Anthropology is the study of people. Ethnography is the practice of immersing oneself in a culture in order to describe that culture. User and task analysis has borrowed from ethnography the methods of both unobtrusive observation (sitting quietly in the corner or like a fly on the wall) and participatory observation (learning about a culture by becoming part of it). Much of the philosophy of user and task analysis derives from ethnography, including respect for the people you are observing and the importance of paying attention to their language, their ways of working, and their environment and culture. It is from ethnography that we realize the importance of seeing who the people are and what they do in their own contexts (their workplace or home). User and task analysis, like ethnography, is about developing as rich an understanding as possible and,

therefore, shares with ethnography the use of qualitative methods of analysis rather than quantitative.

A typical user and task analysis is, of course, not real ethnography. Ethnography usually means spending a year or two immersed in a different environment. Half a day with each user and a few weeks for the entire visiting time isn't the same. Ethnography, moreover, is usually only about describing the culture. A user and task analysis has a goal that goes beyond description. The point of user and task analysis is to design a product that may in fact change the culture being observed. So despite the titles of some papers in the field, we aren't really doing an ethnographic study of a set of users when we do user and task analysis. But we are making practical use of ethnographic philosophy and techniques, adapted to the goals of designing products and the constraints of product schedules and budgets.

Cognitive psychology

Cognitive psychology is the study of how people think and learn. It is from work in cognitive psychology over the last several decades that we have come to appreciate that we cannot just impose designs on users. People are active parts of the system, and because they are much less predictable and less well understood than the computers and other technological parts of the system, they require even greater study and understanding. Users come to any new product with preconceived ideas based on their prior experiences. They interpret what they see in an interface and draw their own conclusions about how it works that may be very different from the designers' intentions—and then they act on their conclusions, not on the designers' intentions. Cognitive psychology shows us that we must accept the users as reality because it is they and not the designers (nor their supervisors) who will in the end determine how the product is used (or not used).

Many of the techniques in user and task analysis and in other usability studies are also adapted from work in cognitive psychology. Think aloud protocols (asking users to talk aloud while working) are a cognitive psychology technique for understanding how users go about working, what they are thinking, and what hypotheses they come up with when things go wrong. And many of the techniques of usability testing derive from work in academic psychology. Once again, however, we have adapted these techniques to the practical realities of projects where the main goal is to design a product, not to do an academic experiment. Even when we are gathering quantitative data in usability studies, it is not to satisfy questions of statistical significance but to convince ourselves and others that we have discovered a problem that requires fixing. Experience has shown that with iterative small-scale usability evaluations, designers can meet the usability goals for products. That's a different use of the technique than is practiced in experimental cognitive psychology.

Document design and technical communication; English and rhetoric

Rhetoric is a discipline usually housed in English departments and is the background of many technical communicators. As Schriver (1997, 58) explains, classical rhetoric (in ancient Greece) was the art of persuasion. Today, it is the art of communicating with others through any medium. Schriver says that the three key ideas that rhetoric brings to practice in writing and graphic design (and thus to design of interfaces, too) are audience, invention, and heuristics. We can reinterpret that as understanding who you are communicating with (users), figuring out how to communicate with them (design), and having guidelines for doing so (heuristics).

Thus user and task analysis shares a focus on communication and users with rhetoric. However, in rhetorical studies, and in most English classes, as in most software and hardware development projects, writers (developers) don't actually go out and meet their audiences. They imagine them. They try to *think* of the users' relevant characteristics, using their own prior knowledge and analytic judgment. In that, user and task analysis in this book differs from traditional rhetoric and traditional software and hardware development. User and task analysis must be empirical, based on actual observations of actual users. Otherwise the chances are too high that the imagined audience will be just like the writer or developer.

Technical communication was done originally by engineers or scientists writing technical documents. Audience, but again usually an imagined audience, has always been part of the technical communication tradition. Then, in the late 1970s and 1980s, the discipline of document design developed, drawing on both traditional rhetoric and traditional technical communication as well as cognitive psychology and instructional systems design. The difference was that document design, as developed and practiced by Ginny and colleagues at the Document Design Center of the American Institutes for Research, as well as by JoAnn at Comtech Services and by Karen Schriver and others at Carnegie-Mellon University, was practical and empirical. Today, technical communication is primarily focused on documentation for software and hardware and follows the practical, empirical techniques of the document design methodology. And that starts with exactly the type of user and task analysis that this book is all about.

Instructional systems design

Instructional systems design (ISD) is a method for developing training. The methodology begins with doing needs analysis, understanding users and what they know, understanding the tasks they must learn to do, judging the gap between what they know and what they do not know, and then designing training to bridge that gap. Thus, user and task analysis has a long history in ISD, and ISD

has great applicability to user and task analysis for new and improved product design.

However, there are differences in approach that both instructional designers and interface designers should understand and appreciate. ISD usually comes into play after the product for which training is needed has been designed. That means the procedures have already been developed and the understanding that instructional designers are seeking is how to get people who do not know the procedures to learn them. Thus, when they do needs analysis, they are looking at novices in the new system to see the need and at experts in the new system to see what the tasks are and how to do them.

This book is about doing user and task analysis when the procedures are not yet known. The goal of predesign user and task analysis is to figure out what procedures are needed and what they should look like. Thus it is much less focused than a typical ISD user and task analysis. The user part may be similar: to find out who is out there, what they know, and how they learn and work. But the task part is not similar. Instead of a neatly known task, in a predesign task analysis, the goal is to see the messy reality of how people really do work, with all the errors and problems that they have. So while the technique of drawing task flowcharts for procedures may be borrowed from ISD, in the type of task analysis we are discussing in this book, the flowcharts won't show just the one best way to do the task that you would want to teach someone. They show all the workarounds and ways that users actually do the tasks today.

Market research

Market research is about studying people as customers and consumers, especially their own views of their needs and desires, their preferences, and their reactions to new ideas. Marketing departments often have very useful information to aid in a user and task analysis. They may have data on size and composition of market segments; they may have demographic information about users as a group; they may have information on specific clients' installations. They may have excellent contacts, and individuals in marketing may know a lot about users and environments because they spend time in the field.

Market research shares the emphasis on people rather than technology that is the main theme of user and task analysis. However, the techniques for user and task analysis that we urge you to adopt differ from some of the more traditional techniques used in market research. Where market research tends to focus on attitudes and opinions, user and task analysis for interface design focuses on behavior. What users say they need may not be the best solution to the problems that are generating the statement of need. (Users may say they need faster computers when the problem is really network response time. The users, or the person representing the users in a focus group, may know only that the work goes

more slowly than they want it to. A skilled observer watching the user working might see the real problem.)

Market research also tends to focus on customers or user representatives, not necessarily on the people who actually do the work. In doing user and task analysis for interface design, you must focus on the users themselves because they are the people who will actually interact with the design. You may also want to talk with supervisors and others, but only in addition to working with the users themselves.

Scientific management

User and task analysis is also indebted to the methods developed at the beginning of the 20th century by advocates of scientific management. Scientific management was a way to increase efficiency of work by doing detailed time-and-motion studies of the steps in a task and then reorganizing the task to save steps and time. The techniques of observing and timing people as they work, describing the tasks they do in detailed steps, and drawing elaborate flowcharts and other graphic presentations of task analyses come from this discipline.

Scientific management, however, was criticized for treating the human as a cog in a machine and not considering all the cognitive as well as physical steps that people take when they do work. Over the years, task analysis has moved beyond scientific management to acknowledge and include the decisions that users make as they work, the knowledge they bring to their tasks, and the ways they use that knowledge. Thus, as with the other disciplines that we have described in this section, the user and task analysis that we focus on in this book is informed by scientific management but moves beyond it in new directions.

Participatory design in the Scandinavian model

We close this chapter with one more antecedent, not a specific discipline, but a tradition of practice from which we draw both philosophy and technique. This is the Scandinavian experiences with participatory analysis and design. [Ehn (1993) is a good place to start for an overview.] For two decades, in Scandinavia, designers and workers have collaborated on understanding users and their tasks and on planning and designing new business practices and interfaces. Because of the power of workers' unions and legal requirements that workers be involved in decisions that will affect them, designers in Scandinavia have developed ways to encourage and realize collaborations between users and designers. Both the philosophy and techniques have influenced analysis and design ideas and methods outside of Scandinavia, including many that we discuss in this book.

References cited in the chapter

Beyer, Hugh and Holtzblatt, Karen, *Contextual Design: Defining Customer-Centered Systems*, San Francisco, CA: Morgan Kaufmann Publishers, 1997.

Dayton, Tom, McFarland, Al, and Kramer, Joseph, Bridging user needs to object oriented GUI prototype via task object design, in *User Interface Design: Bridging the Gap from User Requirements to Design*, edited by Larry E. Wood, Boca Raton, FL: CRC Press, 1998, 15–56.

Ehn, Pelle, Scandinavian design: On participation and skill, in *Participatory Design: Principles and Practices*, edited by Douglas Schuler and Aki Namioka, Hillsdale, NJ: Lawrence Erlbaum Associates, 1993, 41–77.

Hackos, JoAnn T., Finding out what users need and giving it to them: A case-study at Federal Express, *Technical Communication*, 42 (2), 1995: 322–327.

Hiser Group, *The Element Tool Kit*, Prahan, Victoria, Australia, 1997.

Katel, Peter, Bordering on chaos, *Wired Magazine*, July 1997: 98–107.

Landauer, Thomas, *The Trouble with Computers*, Cambridge, MA: MIT Press, 1995.

Schriver, Karen A., *Dynamics in Document Design*, NY: John Wiley & Sons, 1997.

Wixon, Dennis and Jones, Sandy, Usability for fun and profit: A case study of the design of DEC RALLY version 2, in *Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context*, edited by Marianne Rudisill, Clayton Lewis, Peter B. Polson, Timothy D. McKay, San Francisco, CA: Morgan Kaufmann Publishers, 1996, 3–35.

Other books and articles for further reading

Jonassen, David H, Hannum, Wallace H., and Tessmer, Martin, *Handbook of Task Analysis Procedures*, NY: Praeger, 1989.

Kirwan, Barry and Ainsworth, Les K., *A Guide to Task Analysis*, London: Taylor & Francis, 1992.

Moore, Geoffrey, *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers*, NY: Harper Business, 1995.

Norman, Donald A., *The Design of Everyday Things*, NY: Doubleday, 1988 (originally published as *The Psychology of Everyday Things*, hard cover published by Basic Books).

Rubinstein, Richard and Hersh, Harry, *The Human Factor: Designing Computer Systems for People*, Bedford, MA: Digital Equipment Corporation, 1984.

Shneiderman, Ben, *Designing the User Interface: Strategies for Human-Computer Interaction*, 3rd ed., Reading, MA: Addison-Wesley, 1998.

Wixon, Dennis and Ramey, Judith, Eds., *Field Methods Casebook for Software Design*, NY: John Wiley & Sons, 1996.

Wood, Larry E., The ethnographic interview in user-centered work/task analysis, in *Field Methods Casebook for Software Design*, edited by Dennis Wixon and Judith Ramey, NY: John Wiley & Sons, 1996, 35–56.

Zemke, Ron and Kramlinger, Tom, *Figuring Things Out: A Trainer's Guide to Needs and Task Analysis*, Reading, MA: Addison-Wesley, 1982.